

ご購入はこちら

# オープンソースCPU 「RISC-V」の研究

第4回 RISC-Vプログラムを作って命令セット・シミュレータ上で動かす

@msyksphinz

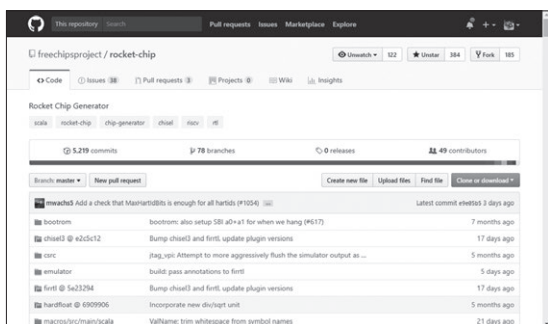


図1 代表的なRISC-V実装であるRocket ChipはGitHubから入手できる

<https://github.com/freechipsproject/rocket-chip>

## 今回やること…まずはボードなしでPC上で試してみる

RISC-Vを試すには幾つか方法があります。

- VerilogシミュレーションやFPGAでの動作確認ができる「Rocket Chip」
- RISC-Vコアが入ったASICボード「HiFive1」

どちらも良いですが、まずは実機を持っていないと始められる、

- ソフトウェア環境の準備
- プログラムのコンパイル
- シミュレーション方法

を試してみましよう。HiFive1ボードなどの実機を持っていないとRISC-Vにトライすることのできる方法を探ります。今回は、RISC-V実装の代表格であるRocket Chipの環境を使います。

### ● 代表的RISC-V実装「Rocket Chip」&開発環境の入手

Rocket Chipのプロジェクトは、GitHubの以下のページからダウンロードできます(図1)。

<https://github.com/freechipsproject/rocket-chip>

<https://github.com/riscv>

<https://github.com/ucb-bar/>

これらのページからは、

- RISC-Vのハードウェア実装であるRocket Chip
- アウト・オブ・オーダーのRISC-VプロセッサであるBOOM (Berkeley Out-of-Order Machine)
- これらのハードウェアを記述するために開発された高位合成言語Chisel
- RISC-VのISAのテスト・パターンである riscv-tests
- コンパイラ riscv-gcc
- シミュレータである riscv-isa-sim など

も一緒にダウンロードできます。まずは一式ダウンロードしてしまうのがよいでしょう(図2)。

### ▶ 必要なディスク空き容量

ちなみに、上記のツール群も一緒にダウンロードして一式シミュレーションし、FPGAに書き込むために論理合成まで行くとすると結構なディスク容量を消費します。少なくとも40G~50Gバイト程度のディスクの空き容量を確保しておくことをお勧めします。

### ● いろいろ試す人向けのダウンロード・テクニク

以降で紹介するRocket Chipのリポジトリや、fpga-zynqやSiFiveの提供するリポジトリには、サブリポジトリとして必ずと言ってよいほどコンパイル・シミュレーション環境を構築するためにriscv-toolsリポジトリが含まれています。一度GCCを用意すれば十分な場合がほとんどなので注1、リポジトリをcloneするごとにriscv-toolsサブリポジトリもダウンロードす

注1: それ以外の場合もあります。例えばライブラリを32ビット版で用意したり、Compact命令込みのGCCツールチェーンを生成したりする場合など。

注: 本稿の内容は執筆時点のもので、随時更新されていく可能性があります。